

.....

```
MM      MM      000000      UU      UU      DDDDDDDD      EEEEEEEEEEE      FFFFFFFFFFF
MM      MM      000000      UU      UU      DDDDDDDD      EEEEEEEEEEE      FFFFFFFFFFF
MMM     MMM     00      00      UU      UU      DD      DD      EE      FF
MMM     MMM     00      00      UU      UU      DD      DD      EE      FF
MM      MM      00      00      UU      UU      DD      DD      EE      FF
MM      MM      00      00      UU      UU      DD      DD      EEEEEEEE      FFFFFFFF
MM      MM      00      00      UU      UU      DD      DD      EEEEEEEE      FFFFFFFF
MM      MM      00      00      UU      UU      DD      DD      EE      FF
MM      MM      00      00      UU      UU      DD      DD      EE      FF
MM      MM      00      00      UU      UU      DD      DD      EE      FF
MM      MM      00      00      UU      UU      DD      DD      EE      FF
MM      MM      000000      UUUUUUUUUU      DDDDDDDD      EEEEEEEEEEE      FF
MM      MM      000000      UUUUUUUUUU      DDDDDDDD      EEEEEEEEEEE      FF
```

....
....
....
....

```
BBBBBBBB      333333      222222
BBBBBBBB      333333      222222
BB      BB      33      33      22      22
BB      BB      33      33      22      22
BB      BB      33      33      22      22
BBBBBBBB      33      22
BBBBBBBB      33      22
BB      BB      33      22
BB      BB      33      22
BB      BB      33      22
BBBBBBBB      333333      2222222222
BBBBBBBB      333333      2222222222
```

Require file for MOUNT utility

Version: 'V04-000'

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

++

FACILITY: MOUNT Utility Structure Level 1

ABSTRACT:

This file contains definitions common to the modules that make up the MOUNT utility.

ENVIRONMENT:

STARLET operating system, including privileged system services and internal exec routines.

--

AUTHOR: Andrew C. Goldstein, CREATION DATE: 29-Sep-1977 16:27

MODIFIED BY:

V03-012 DAS0001 David Solomon 09-Jul-1984
Add /[NO]REBUILD qualifier.
Add VC_SBMSEQNUM field to volume context value block.

V03-011 HH0017 Hai Huang 03-May-1984
Add BEGIN-END block around SURFACE_ERROR macro-body.

V03-010 ACG0408 Andrew C. Goldstein, 22-Mar-1984 21:56
Make APPLY_RVN and DEFAULT_RVN macros

V03-009 HH0004 Hai Huang 09-Mar-1984
Add /CLUSTER qualifier for cluster-wide mount support.

V03-008 MCN0136 Maria del C. Nasr 31-Oct-1983
Eliminate list of qualifiers for DISMOUNT, since they
are no longer needed with the new CLI interface.

V03-007 CDS0001 Christian D. Saether 3-Aug-1983
Define fields within the device lock value block.
Define bit in stored_context field for xqp.

V03-006 MMD0187 Meg Dumont, 7-Jul-1983 9:59
Make the default for AVL/AVR the same from the DCL call
and from the system service call.

V03-005 MMD0112 Meg Dumont, 29-Mar-1983 0:20
Added bit definitions for new qualifiers on mount
for AVL, AVR and new VMS tape protections.

V03-004 SJT50311 Steven T. Jeffreys 25-Feb-1983
Moved STORED_CONTEXT bit definitions to MOUDEF to
prevent a proliferation of local definitions.

V03-003 STJ0316 Steven T. Jeffreys 15-Aug-1982
Add support for journaling qualifiers.

V03-002 STJ0305 Steven T. Jeffreys 18-May-1982
Replace the obsolete OPT_UNLOCK definition with the
OPT_NOUNLOAD definition.

V03-001 STJ0245 Steven T. Jeffreys 04-Apr-1982
- Add TIMER_EFN and MOUNT_EFN definitions.
- Add DO_IO keyword macro.

V02-017 STJ0155 Steven T. Jeffreys 02-Jan-1982
Added OPT_NOCACHE, OPT_OVR_LOCK, OPT_MESSAGE, OPT_MOUNTVER
bit definitions.

V02-016 STJ0016 Steven T. Jeffreys 30-Aug-1981
Added OPT_SIGNAL bit definitions.

V02-015 STJ0086 Steven T. Jeffreys 05-Aug-1981
Changed ERR_EXIT macro to use LIB\$STOP.

V02-014 STJ0035 Steven T. Jeffreys 11-May-1981
Added support for the /ASSIST qualifier.

V02-013 STJ0006 Steven T. Jeffreys 08-Apr-1981
Change ERR_EXIT to SIGNAL rather than SIGNAL_STOP.
Necessary for operator assisted mount.

V02-012 ACG35282 Andrew C. Goldstein, 23-Jan-1981 14:12

Clean up SCB after ACP start failure

V02-011 ACG0175 Andrew C. Goldstein, 23-May-1980 14:39
Add I/O database lock cleanup flag

V02-010 ACG0167 Andrew C. Goldstein, 18-Apr-1980 13:38
Previous revision history moved to MOUNT.REV

**

Definitions of the option bits produced by the command parser.

LITERAL

```

SEQULST (OPT,GBL,0,1
.(DENSITY,)      /DENSITY switch seen
.(DENS_800,)     /DENSITY:800 switch seen
.(VID,)         /VISUAL_IDENTIFICATION switch seen
.(COMMENT,)     /COMMENT switch seen
.(NOSHARE,)     /NOSHARE switch seen
.(EXPSHARE,)    /SHARE or /NOSHARE seen explicitly
.(SHARE,)      /SHARE switch seen
.(GROUP,)      /GROUP switch seen
.(SYSTEM,)     /SYSTEM switch seen
.(WRITE,)      /WRITE switch seen
.(NOUNLOAD,)   /NOUNLOAD switch seen
.(FOREIGN,)    /FOREIGN switch seen
.(NOLABEL,)    /NOLABEL switch seen
.(EBC_DATA,)   /EBCDIC:DATA seen
.(EBC_LABELS,) /EBCDIC:LABELS seen
.(BLOCK,)      /BLOCK switch seen
.(BLOCKSIZE,)  block size specified
.(PROTECTION,) /PROTECTION switch seen
.(OWNER_UIC,)  /OWNER UIC switch seen
.(USER_OIC,)   /USER OIC switch seen
.(OVR_EXP,)    /OVERRIDE:EXPIRATION switch seen
.(OVR_SETID,)  /OVERRIDE:SETID switch seen
.(OVR_ID,)     /OVERRIDE:IDENTIFICATION switch seen
.(EXTENSION,)  /EXTENSION switch seen
.(WINDOW,)     /WINDOW switch seen
.(ACCESSED,)   /ACCESSED switch seen
.(UNIQUEACP,)  /PROCESSOR:UNIQUE switch seen
.(SAMEACP,)    /PROCESSOR:SAME switch seen
.(FILEACP,)    /PROCESSOR:filename switch seen
.(LOG_NAME,)   logical name specified
.(DEVICE,)     device name seen
.(LABEL,)      volume label seen
.(OVR_PRO,)    user is overriding volume protection
.(IS_FILES11,) volume is found to be Files-11
.(IS_FILES11B,) volume is found to be Files-11 Structure level 2
.(READCHECK,)  /DATACHECK:READ switch seen
.(WRITECHECK,) /DATACHECK:WRITE switch seen
.(RECORDSZ,)   /RECORD:value switch seen
.(OVR_ACC,)    /OVERRIDE:ACCESSIBILITY
.(DRIVES,)     /DRIVES:value switch seen
.(BIND,)       /BIND:structure_name switch seen

```

: R


```

,(DO BIND,)      | volume must be bound into set
,(NOQUOTA,)      | /NOQUOTA switch seen
,(DENS 1600,)    | /DENSITY:1600 switch seen
,(NOHDR3,)       | /NOHDR3 switch seen
,(CACHE,)        | /CACHE=(EXTENT:n,FILE_ID:n,QUOTA:n) seen
,(WTHRU,)        | /CACHE=WRITETHROUGH seen
,(NOEXT-C,)      | /CACHE=NOEXTENT seen
,(NOFILE-ID,)    | /CACHE=NOFILE_ID seen
,(NOQUO-C,)      | /CACHE=NOQUOTA seen
,(ASSIST,)       | /ASSIST seen
,(MESSAGE,)      | /MESSAGE seen
,(NOCACHE,)       | /NOCACHE seen
,(OVR LOCK,)     | /OVERRIDE=LOCK seen
,(MOUNTVER,)     | /MOUNT VERIFICATION seen
,(NOJRNAL,)      | /NOJOURNAL seen
,(NEWJRNAL,)     | /JOURNAL=(NEWFILE) seen
,(NOAUTO,)       | /NOAUTOMATIC switch seen
,(INIT-ALL,)     | /INITIALIZE=ALL switch seen
,(INIT-CONT,)    | /INITIALIZE=CONTINUATION switch seen
,(OVR VOLO,)     | /OVERRIDE:OWNER_IDENTIFIER switch seen
,(INTERCHG,)     | /INTERCHANGE switch seen
,(CLUSTER,)      | /CLUSTER switch seen
,(NOREBUILD,)    | /NOREBUILD switch seen
);

```

Masks of options legal for disk and tape.

LITERAL

```

DISK_OPTIONS1 = | qualifiers legal for disk
1^OPT_VID       | /VISUAL_IDENTIFICATION switch seen
OR 1^OPT_COMMENT | /COMMENT switch seen
OR 1^OPT_NOSHARE | /NOSHARE switch seen
OR 1^OPT_EXPSHARE | /SHARE or /NOSHARE seen explicitly
OR 1^OPT_SHARE   | /SHARE switch seen
OR 1^OPT_GROUP   | /GROUP switch seen
OR 1^OPT_SYSTEM  | /SYSTEM switch seen
OR 1^OPT_WRITE   | /WRITE switch seen
OR 1^OPT_NOUNLOAD | /NOUNLOAD switch seen
OR 1^OPT_FOREIGN | /FOREIGN switch seen
OR 1^OPT_BLOCK   | /BLOCK switch seen
OR 1^OPT_PROTECTION | /PROTECTION switch seen
OR 1^OPT_OWNER_UIC | /OWNER_UIC switch seen
OR 1^OPT_USER_OIC | /USER_OIC switch seen
OR 1^OPT_OVR_ID   | /OVERRIDE:IDENTIFICATION switch seen
OR 1^OPT_EXTENSION | /EXTENSION switch seen
OR 1^OPT_WINDOW   | /WINDOW switch seen
OR 1^OPT_ACCESSED | /ACCESSED switch seen
OR 1^OPT_UNIQUEACP | /PROCESSOR:UNIQUE switch seen
OR 1^OPT_SAMEACP  | /PROCESSOR:SAME switch seen
OR 1^OPT_FILEACP  | /PROCESSOR:filename switch seen
OR 1^OPT_LOG_NAME | logical name specified
OR 1^OPT_DEVICE   | device name seen
OR 1^OPT_LABEL    | volume label seen

DISK_OPTIONS2 = | qualifiers legal for disk, second long word
1*(OPT_READCHECK-32) | /DATACHECK:READ switch seen

```

```

OR 1^(OPT_WRITECHECK-32) ! /DATACHECK:WRITE switch seen
OR 1^(OPT_BIND-32)       /BIND:structure name switch seen
OR 1^(OPT_DO_BIND-32)    volume must be bound into set
OR 1^(OPT_NOQUOTA-32)    /NOQUOTA switch seen
OR 1^(OPT_CACHE-32)      /CACHE=(EXTENT:n,FILE_ID:n,QUOTA:n)seen
OR 1^(OPT_WTHRU-32)      /CACHE=WRITETHROUGH seen
OR 1^(OPT_NOEXT_C-32)    /CACHE=NOEXTENT seen
OR 1^(OPT_NOFILE_ID-32)  /CACHE=NOFILE_ID seen
OR 1^(OPT_NOQUO_C-32)    /CACHE=NOQUOTA seen
OR 1^(OPT_ASSIST-32)     /ASSIST seen
OR 1^(OPT_MESSAGE-32)    /MESSAGE seen
OR 1^(OPT_NOCACHE-32)    /NOCACHE seen
OR 1^(OPT_OVR_LOCK-32)   /OVERRIDE=LOCK seen
OR 1^(OPT_NOJRN-32)      /NOJOURNAL seen
OR 1^(OPT_NEWJRN-32)     /JOURNAL=(NEWFILE) seen
OR 1^(OPT_MOUNTVER-32)   /MOUNT VERIFICATION seen
OR 1^(OPT_NOAUTO-32)     /NOAUTOMATIC seen
OR 1^(OPT_CLUSTER-32)    /CLUSTER seen
OR 1^(OPT_NOREBUILD-32), /NOREBUILD seen

```

```

TAPE_OPTIONS1 =
1^OPT_DENSITY           qualifiers legal for tape
OR 1^OPT_DENS_800       /DENSITY switch seen
OR 1^OPT_VID            /DENSITY:800 switch seen
OR 1^OPT_COMMENT        /VISUAL IDENTIFICATION switch seen
OR 1^OPT_NOSHARE        /COMMENT switch seen
OR 1^OPT_EXPSHARE       /NOSHARE switch seen
OR 1^OPT_WRITE          /SHARE or /NOSHARE seen explicitly
OR 1^OPT_NOUNLOAD       /WRITE switch seen
OR 1^OPT_FOREIGN        /NOUNLOAD switch seen
OR 1^OPT_NOLABEL        /FOREIGN switch seen
OR 1^OPT_EBC_DATA       /NOLABEL switch seen
OR 1^OPT_EBC_LABELS     /EBCDIC:DATA seen
OR 1^OPT_BLOCK          /EBCDIC:LABELS seen
OR 1^OPT_BLOCKSIZE      /BLOCK switch seen
OR 1^OPT_PROTECTION     block size specified
OR 1^OPT_OWNER UIC      /PROTECTION switch seen
OR 1^OPT_USER OIC       /OWNER UIC switch seen
OR 1^OPT_OVR_EXP        /USER OIC switch seen
OR 1^OPT_OVR_SETID      /OVERRIDE:EXPIRATION switch seen
OR 1^OPT_OVR_ID         /OVERRIDE:SETID switch seen
OR 1^OPT_OVR_VOLO       /OVERRIDE:IDENTIFICATION switch seen
OR 1^OPT_UNIQUEACP      /OVERRIDE:OWNER IDENTIFIER switch seen
OR 1^OPT_SAMEACP        /PROCESSOR:UNIQUE switch seen
OR 1^OPT_FILEACP        /PROCESSOR:SAME switch seen
OR 1^OPT_LOG_NAME       /PROCESSOR:filename switch seen
OR 1^OPT_DEVICE         logical name specified
OR 1^OPT_LABEL          device name seen
                        volume label seen

```

```

TAPE_OPTIONS2 =
1^(OPT_READCHECK-32)    qualifiers legal for tape, second long word
OR 1^(OPT_WRITECHECK-32) /DATACHECK:READ switch seen
OR 1^(OPT_RECORDS2-32)  /DATACHECK:WRITE switch seen
OR 1^(OPT_DENS_1600-32) /RECORD: value switch seen
OR 1^(OPT_NOHDR3-32)    /DENSITY:1600 switch seen
OR 1^(OPT_OVR_ACC-32)   /NOHDR3 switch seen
                        /OVERRIDE:ACCESSIBILITY switch seen

```



```

:      OR 1^(OPT_DRIVES-32)      : /DRIVES:value switch seen
:      OR 1^(OPT_ASSIST-32)      : /ASSIST seen
:      OR 1^(OPT_MOUNTVER-32)    : /MOUNT VERIFICATION seen
:      OR 1^(OPT_MESSAGE-32)     : /MESSAGE seen
:      OR 1^(OPT_NOAUTO-32)      : /NOAUTOMATIC seen
:      OR 1^(OPT_INIT_ALL-32)    : /INITIALIZE=ALL seen
:      OR 1^(OPT_INIT_CONT-32)   : /INITIALIZE=CONTINUATION seen
:      OR 1^(OPT_OVR_VOLO-32)    : /OVERRIDE:OWNER_IDENTIFIER seen
:      OR 1^(OPT_INTERCHG-32);   : /INTERCHANGE seen

```

```

: Masks of option bits to reset at end of disk mount loop.
:

```

```

LITERAL
  RESET_OPTIONS1 =
    0              ! none in first word
  RESET_OPTIONS2 =
    1^(OPT_OVR_PRO-32)
  OR 1^(OPT_IS_FILES11-32)
  OR 1^(OPT_IS_FILES11B-32)
  OR 1^(OPT_DO_BIND-32);

```

```

: Status bits used for exit processing.
:

```

```

LITERAL
  CLF_DEALLOCATE = 1,      ! DEALLOCATE DEVICE
  CLF_DELAQB     = 2,      ! delete AQB
  CLF_DEASSMBX   = 3,      ! deassign mailbox
  CLF_DEASSTEMP  = 4,      ! deassign temporary channel
  CLF_RELOCK     = 5,      ! restore index file write locks
  CLF_DISMOUNT   = 6,      ! do full dismount to clean up
  CLF_CLEARVALID = 7,      ! clear volume valid bit on failure
  CLF_LOCKVOLUME = 8,      ! leave volume allocation locked
  CLF_REBUILD    = 9,      ! rebuild volume bitmaps
  CLF_REBUILDQUO = 10,     ! rebuild volume set quota file
  CLF_UNLOCKDB   = 11,     ! unlock I/O database
  CLF_CLEANSGB   = 12;     ! clear dirty bits in SCB

```

```

: MAXIMUM VARIABLES
:

```

```

LITERAL
  DEVMAX = 16,      !MAXIMUM NUMBER OF DEVICES IN COMMAND
  LABMAX = 16;      !MAXIMUM NUMBER OF LABELS IN COMMAND

```

```

: DEFINE LENGTH OF NAME STRING BUFFER
:

```

```

LITERAL
  NAMEBUF_LEN = 32;      !LENGTH OF NAME STRING BUFFER

```

```

: Set processor IPL
:

```

```

MACRO
  SET_IPL (LEVEL) =
    BEGIN

```



```

BUILTIN MTPR;
MTPR (%REF (LEVEL), PR$_IPL)
END
%;
```

```

: Error exit macro.
```

```

MACRO
ERR_EXIT (CODE) =
    SIGNAL_STOP (%IF NOT %NULL (CODE) %THEN CODE %ELSE 0 %FI
                %IF NOT %NULL (%REMAINING) %THEN , %REMAINING %FI)
%,
ERR_MESSAGE [] =
    SIGNAL (%REMAINING)
%;
```

```

: Macro used to signal fatal errors (internal consistency checks).
```

```

MACRO
BUG_CHECK (CODE, TYPE, MESSAGE) =
    BEGIN
    BUILTIN BUGW;
    EXTERNAL LITERAL %NAME('BUG$_', CODE);
    BUGW (%NAME('BUG$_', CODE) OR 4)
    END
%;
```

```

: Macro to call the change mode to kernel system service.
: Macro call format is 'KERNEL_CALL (ROUTINE, ARG1, ARG2, ... )'.
```

```

MACRO
KERNEL_CALL (R) =
    BEGIN
    EXTERNAL ROUTINE SYSSCMKRNL : ADDRESSING_MODE (ABSOLUTE);
    BUILTIN SP;
    SYSSCMKRNL (R, .SP, %LENGTH-1
                %IF %LENGTH GTR 1 %THEN , %REMAINING %FI)
    END%;
```

```

: Macro to call the change mode to exec system service.
: Macro call format is 'EXEC_CALL (ROUTINE, ARG1, ARG2, ... )'.
```

```

MACRO
EXEC_CALL (R) =
    BEGIN
    EXTERNAL ROUTINE SYSSCMEXEC : ADDRESSING_MODE (ABSOLUTE);
    BUILTIN SP;
    SYSSCMEXEC (R, .SP, %LENGTH-1
                %IF %LENGTH GTR 1 %THEN , %REMAINING %FI)
    END%;
```

```

: Macro to generate a string and descriptor.
```

```

MACRO
```

```
DESCRIPTOR (STRING) =  
  UPLIT (%CHARCOUNT (STRING), UPLIT BYTE (STRING))%;
```

```
! Macro to declare parameter list for TPARSE action routines.
```

```
MACRO
```

```
  TPARSE_ARGS (NAME) =  
    BUILTIN AP;  
    BIND NAME = AP : REF BBLOCK;  
    %;
```

```
! Structure for all SDL defined blocks.
```

```
STRUCTURE
```

```
  BBLOCK [O, P, S, E; N] =  
    [N]  
    (BBLOCK+O)<P,S,E>,  
  
  BBLOCKVECTOR [I, O, P, S, E; N, BS] =  
    [N*BS]  
    ((BBLOCKVECTOR+I*BS)+O)<P,S,E>;
```

```
! Macros to apply the current RVN to a file ID from the file structure,  
! and default the RVN to zero when it is the current one.
```

```
MACRO
```

```
  APPLY_RVN (RVN, CURRENT_RVN) =  
    BEGIN  
    IF .(RVN)<0,8> EQL 0  
    THEN (RVN)<0,8> = CURRENT_RVN;  
    IF .(RVN)<0,8> EQL 1  
    AND CURRENT_RVN EQL 0  
    THEN (RVN)<0,8> = 0;  
    END  
    %;
```

```
  DEFAULT_RVN (RVN, CURRENT_RVN) =  
    BEGIN  
    IF .(RVN)<0,8> EQL CURRENT_RVN  
    THEN (RVN)<0,8> = 0;  
    END  
    %;
```

```
! Macro to evaluate a disk error status code as being a surface error  
! (i.e., caused by the disk medium as opposed to the controller).
```

```
MACRO
```

```
  SURFACE_ERROR (CODE) =  
    BEGIN  
    CODE EQL SSS_PARITY  
    OR CODE EQL SSS_DATACHECK  
    OR CODE EQL SSS_FORMAT  
    OR CODE EQL SSS_FORCEDERROR
```


END
%;

Define a symbol to represent the common event flag that will be specified (if necessary) for any I/O done within the \$MOUNT and \$DISMOU system services.

LITERAL

```
TIMER_EFN      = 25,      ! A reserved, system event flag
MOUNT_EFN      = 26;      ! A reserved, system event flag
```

Define a macro that will be used by any and all modules in \$MOUNT that want to do synchronous I/O via a direct \$QIO. The macro is a convenient way to provide a \$QIO-like interface to a routine that can tolerate some bizarre (mis)use of the event flags by the user.

KEYWORDMACRO

```
DO_IO (EFN      = MOUNT_EFN,      ! Event flag to set on I/O completion (value)
      CHAN      = 0,              ! Channel to device (value)
      FUNC      = 0,              ! I/O function code (value)
      IOSB      = 0,              ! I/O status block (address)
      ASTADR     = 0,              ! Ast routine (address)
      ASTPRM     = 0,              ! Ast parameter (value)
      P1         = 0,              ! Optional device/function parameter
      P2         = 0,              ! ditto
      P3         = 0,              ! ditto
      P4         = 0,              ! ditto
      P5         = 0,              ! ditto
      P6         = 0,              ! ditto
      ) =
```

BEGIN

EXTERNAL ROUTINE

```
COMMON_IO : ADDRESSING_MODE (GENERAL);
COMMON_IO (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
END
%;
```

Define bitnames for the STORED_CONTEXT bitvector

LITERAL

```
TAPE_MOUNT      = 0,      ! was the first device a tape or disk
OVERRIDE_SOMETHING = 1,    ! is something being overridden
XQP              = 2;      ! XQP is being used.
```

Define fields within the device allocation lock value block.

FIELD

DC =

```
SET
DC_FLAGS          = [0,0,16,0],
DC_NOTFIRST_MNT   = [0,0,1,0],
DC_FOREIGN        = [0,1,1,0],
```

```
DC_GROUP          = [0,2,1,0],
DC_SYSTEM         = [0,3,1,0],
DC_WRITE          = [0,4,1,0],
DC_NOQUOTA        = [0,5,1,0],
DC_OVR_PROT       = [0,6,1,0],
DC_OVR_OWNUIC     = [0,7,1,0],
DC_NOINTERLOCK    = [0,8,1,0],

DC_PROTECTION     = [2,0,16,0],
DC_OWNER_UIC      = [4,0,32,0]
TES;
```

```
! Define fields for volume lock value block.
!
```

```
FIELD
VC =
SET
VC_NOTFIRST_MNT   = [0,0,1,0],
VC_NOALLOC        = [0,1,1,0],
VC_IBMAPVBN       = [2,0,8,0],
VC_SBMAPVBN       = [3,0,8,0],
VC_VOLFREE        = [4,0,32,0],
VC_IDXFILEOF      = [8,0,32,0],
VC_SBMSEQNUM       = [12,0,16,0]
TES;
```


0243 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

